
Visma Documentation

Release 0.1.0

Henrik Palmlund Wahlgren

Oct 29, 2018

Contents:

1	Installation	3
2	Access to Visma API	5
3	Environment Variables	7
4	Using the library	9
4.1	Create object	9
4.2	Get objects	9
4.3	Update object	9
4.4	Delete object	10
4.5	Filter objects	10
4.5.1	Filtering functions	10
4.6	Order by	11
4.7	Get first object	11
5	Models	13
5.1	API Models	13
5.1.1	Account	13
5.1.2	AccountType	14
5.1.3	Article	14
5.1.4	ArticleAccountCoding	15
5.1.5	ArticleLabel	16
5.1.6	Bank	17
5.1.7	BankAccount	17
5.1.8	CompanySettings	18
5.1.9	CostCenter	19
5.1.10	CostCenterItem	19
5.1.11	Country	20
5.1.12	Currency	20
5.1.13	Customer	20
5.1.14	CustomerInvoiceDraft	23
5.1.15	CustomerLabel	25
5.1.16	DeliveryMethod	25
5.1.17	DeliveryTerm	25
5.1.18	FiscalYear	26
5.1.19	ForeignPaymentCodes	26

5.1.20	Project	26
5.1.21	OpeningBalances	27
5.1.22	Supplier	27
5.1.23	Unit	29
5.1.24	User	29
5.1.25	VatCode	29
5.1.26	VatReport	30
5.1.27	TermsOfPayment	30
5.2	Models without endpoints	31
5.2.1	CompanyTexts	31
5.2.2	CustomerInvoiceDraftRow	32
5.2.3	DocumentApprovalEvent	33
5.2.4	PaginatedResponse	33
5.2.5	PaginationMetadata	33
5.2.6	RelatedAccounts	33
6	REST-ORM Framework	35
6.1	Inspiration	35
6.2	Models	35
6.3	Model Meta	36
6.4	Endpoints and methods	36
6.5	Enveloped Schemas	36
7	Commercial Support and Development	39
8	Indices and tables	41
	Python Module Index	43

Visma is a python client/ORM for the Visma eAccounting API

CHAPTER 1

Installation

Using the library requires Python 3.6 or higher

```
pip install visma
```


CHAPTER 2

Access to Visma API

After installation you will need to set up access to the Visma E-Accounting API

As of now it is not possible to get access by yourself so you will need to contact Visma at eaccountingapi@visma.com. When you contact them request redirect URI to `/redirect_receiver`

They will first set up an account in their sandbox environment for you and after you have tested you use cases you can request an account in the Visma Production Environment

The focus of the library as of now is to be used in a command line tool for generating invoices from data from our time reporting system. In the future we might extend it to be a Django App.

Since we are focusing on command line tooling we run in to some “problems” with authentication since Visma is using OAuth2. This requires you to go through a website to get an access code and verify access rights in the Visma API.

We provide an entry point to open the correct webpage

```
visma request_access --client <client_id>
```

This will open a webpage where you log in and grant access to the application. After you click OK you will be redirected to the `/redirect_receiver` url. You need to take the code in the url and use it to feed into the authenticate function

```
visma get_token --code <auth_code> --client <client_id> --secret <client_secret> > /  
↪path/to/auth.json
```

This will return an access token and a renew token that will be used by the client to make authenticated requests to the Visma API.

Environment Variables

To set up the API classes you need to supply give your Visma Client ID, Visma Client Secret and the path to the tokens you saved before.

These settings are now supplied via the environment variables:

- VISMA_API_CLIENT_ID
- VISMA_API_CLIENT_SECRET
- VISMA_API_TOKEN_PATH
- VISMA_API_CLASS

If you are using the test environment supplied from Visma API Team you need to add the environment variable VISMA_API_ENV=test so that the paths are set up properly.

4.1 Create object

You create object as you would normally in Python. To save the data in Visma you need to call the `.save()` method

```
customer = Customer(  
    invoice_city='Helsingborg',  
    invoice_postal_code='25234',  
    name='TestCustomer AB',  
    terms_of_payment_id='8f9a8f7b-5ea9-44c6-9725-9b8a1addb036',  
    is_private_person=False,  
    is_active=True)  
  
customer.save()
```

4.2 Get objects

You can query the object endpoint by going via the manger on the object.

```
customers = Customer.objects.all()
```

If you just want a specific object that you know the primary key for you can use the `.get()`

```
customer = Customer.objects.get('8f9a8f7b-5ea9-44c6-9725-9b8a1addb036')
```

4.3 Update object

When updating an object you just need to get the object, make changes and save.

```
customer = Customer.objects.get('8f9a8f7b-5ea9-44c6-9725-9b8a1addb036')
customer.name = 'New Name'
customer.save()
```

4.4 Delete object

You can issue a delete on a collected object or use the `.delete` on the manager if you know the primary key.

```
customer = Customer.objects.get('8f9a8f7b-5ea9-44c6-9725-9b8a1addb036')
customer.delete()

# or

Customer.objects.delete('8f9a8f7b-5ea9-44c6-9725-9b8a1addb036')
```

4.5 Filter objects

You can filter result on the API-side using `.filter()` and `.exclude()`. They are chainable so you can combine them.

```
# You don't need to use the .all()
customers = Customer.objects.all().filter(name='Dave')
# is the same as:
customers = Customer.objects.filter(name='Dave')

# You can combine filter and exclude
customers = Customer.objects.filter(name='Dave').exclude(customer_number='1337')
    ↳filter(delivery_invoice_country='Sweden')
```

4.5.1 Filtering functions

There are some special filtering functions that can be used. They can be used in both `.filter()` and `.exclude()`. When using filtering functions in `exclude` they are handled as the inverse of what would be included using `.filter()`

exact using `{arg}__exact` is the same as just supplying the arg. Ex `name__exact='Dave'` or `name='Dave'`. It will handle parameters that match exactly.

not using `{arg}__not` will match to anything but the supplied value. (opposite to exact)

greater than using the `{arg}__gt` argname will translate into filter where greater than the supplied value.

greater than or equal using `{args}__gte` will translate into filter where greater than or equal the supplied value

less than using the `{arg}__lt` argname will translate into filter where less than the supplied value.

less than or equal using `{args}__lte` will translate into filter where less than or equal the supplied value

```
customers = Customer.objects.filter(name__not='Dave').exclude(invoice_zip_code__
    ↳gte=27000)

#querysets are lazy and can be changed until it has been evaluated.

customers.filter(customer_number__lt=4).filter(invoice_country='Sweden')
```

4.6 Order by

You can order result. To supply what arg you want the result to be ordered by use `.order_by()`. Ex `.order_by(name)`

```
customers = Customer.objects.all().order_by(name)
```

4.7 Get first object

If you are interested in the first object of a result you can use normal list handling of `result[0]` but you can also use the function `.first()` to make it a bit more clear and so that the queryset returns an object instead of a list of objects.

```
customer = Customer.objects.filter(customer_number=1337).first()
```


Here is the different models you can interact with through the Visma eAccounting API documented.

5.1 API Models

5.1.1 Account

```
class visma.models.Account(*args, **kwargs)
    Represents a Bookkeeping Account in eAccounting
    endpoint /accounts
    allowed_methods ['list', 'create']
    envelopes {'list': {'class': PaginatedResponse, 'data_attr': 'Data'}}
```

Todo: There is more special cases endpoint for this objects that is not handled yet by the framework.

Parameters

- **name** (*str*) – The name of the account Max length: 100 characters
- **number** (*str*) – required The account number
- **vat_code_id** (*uuid.UUID*) – = Reference to the *VatCode* that is associated with the account
- **vat_code_description** (*str*) – read-only Describes the *VatCode* that is associated with the account
- **fiscal_year_id** (*uuid.UUID*) – required Reference to the *FiscalYear* that the account belongs to.

- **reference_code** (*str*) – read-only The reference code on the account. Dutch companies only
- **type** (*int*) – read-only The account type number. Dutch companies only
- **type_description** (*str*) – read-only The account type description. Dutch companies only
- **modified_utc** (*datetime.datetime*) – read_only Modified date.
- **is_active** (*bool*) – required Indicates if the account is active. default=False
- **is_project_allowed** (*bool*) – Indicates if the account can be used for project bookkeeping. default=False
- **is_cost_center_allowed** (*bool*) – Indicates if the account can be used for cost center bookkeeping. default=False
- **is_blocked_for_manual_booking** (*bool*) – Indicates if the account can be used for manual verification registering.

5.1.2 AccountType

class visma.models.**AccountType** (*args, **kwargs)

Default Account Types. This is applicable on all countries but most relevant for the Netherlands

endpoint /accounttypes

allowed_methods ['list']

envelopes {'list': {'class': PaginatedResponse, 'data_attr': 'Data'}}

Parameters

- **type** (*int*) – Account type number.
- **type_description** (*str*) – Account type description

5.1.3 Article

class visma.models.**Article** (*args, **kwargs)

Represents an Article in eAccounting.

endpoint /articles

allowed_methods ['list', 'get', 'create', 'update']

envelopes {'list': {'class': PaginatedResponse, 'data_attr': 'Data'}}

Parameters

- **id** (*uuid.UUID*) – read-only Unique Id provided by eAccounting.
- **is_active** (*bool*) – required Indicates if the article is active. default=True
- **number** (*str*) – required Article number Max length: 40 characters
- **name** (*str*) – required Article name Max length: 50 characters
- **name_english** (*str*) – Article name in english Max length: 50 characters
- **net_price** (*number*) – Net price of article Max Value: 10000000, Format: Max 2 decimals, default=0

- **gross_price** (*number*) – Gross price of article Max Value: 10000000, Format: Max 2 decimals, default=0
- **coding_id** (*uuid.UUID*) – required Reference to [ArticleAccountCoding](#) used for the article.
- **coding_name** (*str*) – read-only Article account coding name.
- **unit_id** (*uuid.UUID*) – required Reference to [Unit](#) used for the article.
- **unit_name** (*str*) – read-only Name of unit specified in unit_id.
- **unit_abbreviation** (*str*) – read-only Unit abbreviation of unit specified in unit_id.
- **stock_balance** (*number*) – Stock balance for the article. default=0
- **stock_balance_manually_changed_utc** (*datetime.datetime*) – read-only Set when the stock balance is changed manually, for example after an inventory check.
- **stock_balance_reserved** (*number*) – read-only The reserved stock balance for the article.
- **stock_balance_available** (*number*) – read-only The available stock balance for the article.
- **changed_utc** (*datetime.datetime*) – Date and time from when the last changes was made on the article
- **house_work_type** (*int*) – House Work Type
- **purchase_price** (*number*) – Purchase price default=0
- **purchase_price_manually_changed_utc** (*datetime.datetime*) – read-only Set when the purchase price is changed manually
- **send_to_webshop** (*bool*) – If True , will send article to VismaWebShop (If company has the integration). default=True
- **article_labels** (*list* ([ArticleLabel](#))) – A list of [ArticleLabel](#)

Todo: work_house_type is not documented. Need to contact Visma API Team.

5.1.4 ArticleAccountCoding

```
class visma.models.ArticleAccountCoding(*args, **kwargs)
    Represents article account coding
    endpoint /articleaccountcodings
    allowed_methods ['list', 'get']
    envelopes {'list': {'class': PaginatedResponse, 'data_attr': 'Data'}}
```

Parameters

- **id** (*uuid.UUID*) – read-only Unique Id provided by eAccounting
- **name** (*string*) – Name
- **name_english** (*string*) – English Name

- **type** (*string*) – Type
- **vat_rate** (*string*) – VAT rate
- **is_active** (*bool*) – Indicates if the Article account coding is active.
- **vat_rate_percent** (*number*) – VAT rate in percentage.
- **domestic_sales_subject_to_reversed_construction_vat_account_number** (*int*) – Account number for domestic sales subject to reversed construction VAT
- **domestic_sales_subject_to_vat_account_number** (*int*) – Account number for domestic sales subject to VAT.
- **domestic_sales_vat_exempt_account_number** (*int*) – Account number for domestic sales that are exempt from VAT.
- **foreign_sales_subject_to_moss_account_number** (*int*) – Account number for foreign sales subject to MOSS.
- **foreign_sales_subject_to_third_party_sales_account_number** (*int*) – Account number for foreign sales subject to third party sales rules.
- **foreign_sales_subject_to_vat_within_eu_account_number** (*int*) – Account number for foreign sales subject to VAT within EU.
- **foreign_sales_vat_exempt_outside_eu_account_number** (*int*) – Account number for foreign sales that are exempt VAT outside of EU.
- **foreign_sales_vat_exempt_within_eu_account_number** (*int*) – Account number for foreign sales that are exempt VAT within EU.
- **domestic_sales_vat_code_exempt_account_number** (*int*) – Account number for domestic sales exempt a VAT Code.
- **changed_utc** (*datetime.datetime*) – read-only Last Changed Time

Todo: It is not yet supported to filter on date for the account.

5.1.5 ArticleLabel

class visma.models.**ArticleLabel** (*args, **kwargs)

Represents an Article Label in eAccounting.

endpoint /articlelabels

allowed_methods ['list', 'create', 'get', 'update', 'delete']

envelopes {'list': {'class': PaginatedResponse, 'data_attr': 'Data'}}

Parameters

- **id** (*uuid.UUID*) – read-only Unique Id provided by eAccounting
- **name** (*str*) – Article label name Max length: 50 characters
- **description** (*str*) – Article label description Max length: 400 characters

5.1.6 Bank

```
class visma.models.Bank (*args, **kwargs)
    Represents a bank in eAccounting

    endpoint /banks

    allowed_methods ['list']

    envelopes {'list': {'class': PaginatedResponse, 'data_attr': 'Data'}}
```

Parameters

- **id** (*uuid.UUID*) – Unique Id provided by eAccounting
- **name** (*str*) – Bank name

5.1.7 BankAccount

```
class visma.models.BankAccount (*args, **kwargs)
    Represents a bank account in eAccounting.

    endpoint /bankaccounts

    allowed_methods ['list', 'create', 'get', 'update', 'delete']

    envelopes {'list': {'class': PaginatedResponse, 'data_attr': 'Data'}}
```

Parameters

- **id** (*uuid.UUID*) – read-only Unique Id provided by eAccounting
- **bank** (*uuid.UUID*) – Reference to a [Bank](#). Not required for bank accounts of cash or tax account type
- **bank_account_type** (*int*) – required 1 = ChequeAccount, 2 = CashAccount, 3 = SavingsAccount, 4 = CurrencyAccount, 5 = DigitalWalletAccount, 6 = CashCreditAccount, 7 = TaxAccount
- **bank_account_type_description** (*str*) – read-only Description of the Bank Account type
- **bban** (*str*) – Bank Account number. Not required for bank accounts of cash or tax account type
- **iban** (*str*) – IBAN number
- **name** (*str*) – Name of Bank account.
- **is_active** (*bool*) – Indicates if the account is active. default=False
- **ledger_account_number** (*int*) – required Account number to do bookkeeping on for the bank account.
- **has_active_bank_agreement** (*bool*) – Indicates if the bank account has an active bank agreement. default=False
- **is_default_cheque_account** (*bool*) – Indicates if the account is the default cheque account. Only used when having several cheque accounts. default=False

5.1.8 CompanySettings

class visma.models.CompanySettings (*args, **kwargs)

The company settings for current user

endpoint /companysettings

allowed_methods ['list', 'update']

Note: As of now there is no special way to handle an endpoint that only contains a singel object accesses without an id. So even if it is a single object you will need to use the following to get the data.

```
>>> company_settings = CompanySettings.objects.all().first()
```

Parameters

- **id** (*uuid.UUID*) – Is set to empty string. This is to allow for updates on the model.
- **name** (*str*) – required Company name.
- **email** (*str*) – Company email.
- **phone** (*str*) – Company phone.
- **mobile_phone** (*str*) – Company mobile phone.
- **address1** (*str*) – Company address, Row 1
- **address2** (*str*) – Company address, Row 2
- **country_code** (*str*) – Country code.
- **postal_code** (*str*) – Company postal code.
- **city** (*str*) – Company city.
- **website** (*str*) – Company website.
- **currency_code** (*str*) – Standard Currency code.
- **terms_of_payment_id** (*uuid.UUID*) – Reference to standard *TermsOfPayment* for company.
- **corporate_identity_number** (*str*) – read-only Corporate Identity Number
- **vat_code** (*str*) – VAT identification number
- **bank_giro** (*str*) – Bankgiro number. Only used in Sweden.
- **plus_giro** (*str*) – Plusgiro number. Only used in Sweden.
- **bank_account** (*str*) – Bank account number.
- **iban** (*str*) – IBAN number.
- **accounting_locked_to** (*datetime.datetime*) – Datetime where the accounting is locked to.
- **gln** (*str*) – Global Location Number.
- **product_variant** (*int*) – read-only Variant of eAccounting. 1 = Standard/Smart, 2 = Invoicing, 3 = Bookkeeping, 4 = Start/Solo, 5 = Pro, 6 = InvoicingCollaboration

- **type_of_business** (*int*) – read-only Indicates the companys business type. 1 = Corporation, 2 = SoleProprietorship, 3 = EconomicAssociation, 4 = NonProfitOrganization, 5 = GeneralPartnership, 6 = LimitedPartnership, 7 = Cooperatives, 9 = PublicLimited
- **vat_period** (*int*) – read-only Period when VAT report should be sent. 1 = OnceAMonth12th, 2 = OnceAMonth26th, 3 = OnceAQuarter, 4 = OnceAYear, 5 = Never, 6 = Bimonthly, 7 = OnceAMonth, 8 = TwiceAYear, 9 = OnceAQuarterFloating
- **activated_modules** (*list(str)*) – List of activated modules.
- **company_text** (*CompanyText*) – A CompanyText: object.
- **next_customer_number** (*int*) – read-only Next customer number in sequence.
- **next_supplier_number** (*int*) – read-only Next supplier number in sequence.
- **next_customer_invoice_number** (*int*) – read-only Next customer invoice number in sequence.
- **next_quote_number** (*int*) – read-only Next quote number in sequence.
- **show_prices_excl_vat_pc** (*bool*) – Indicates if prices should be shown excluding VAT for private individuals/customers.

5.1.9 CostCenter

class visma.models.**CostCenter** (*args, **kwargs)

Represents a cost center. A cost center is a way of splitting up costs and earnings between different parts of the company so it is possible to follow up better.

endpoint /costcenters

allowed_methods ['list', 'update']

envelopes {'list': {'class': PaginatedResponse, 'data_attr': 'Data'}}

Parameters

- **id** (*uuid.UUID*) – read-only Unique Id provided by eAccounting.
- **name** (*str*) – Cost center name.
- **number** (*int*) – Cost center number.
- **is_active** (*bool*) – Indicates if the cost center is active. default=False
- **items** (*list(CostCenterItem)*) – List of *CostCenterItem* belonging to the cost center.

5.1.10 CostCenterItem

class visma.models.**CostCenterItem** (*args, **kwargs)

The actual cost center item in a cost center where the expence gets booked.

endpoint /costcenteritems

allowed_methods 'create', 'get', 'update']

:argument uuid.UUID id = read-only Unique Id provided by eAccounting. :argument uuid.UUID cost_center_id: required Reference to *CostCenter* holding the cost center item. :argument str name: Name of cost center item. Max length: 50 characters :argument str short_name: Short name of cost

center item. Max length: 9 characters :argument bool is_active: Indicates if the cost center item is active.

5.1.11 Country

```
class visma.models.Country(*args, **kwargs)
    Countries available in eAccounting.

    endpoint /countries

    allowed_methods ['list']

    envelopes {'list': {'class': PaginatedResponse, 'data_attr': 'Data'}}
```

Parameters

- **name** (*str*) – Country Name.
- **code** (*str*) – Country Code.
- **is_eu_member** (*bool*) – Indicates if the country is a member of the EU.

5.1.12 Currency

```
class visma.models.Currency(*args, **kwargs)
    Currency available in eAccounting

    endpoint /currencies

    allowed_methods ['list']

    envelopes {'list': {'class': PaginatedResponse, 'data_attr': 'Data'}}
```

Parameters **code** (*str*) – Currency code.

5.1.13 Customer

```
class visma.models.Customer(*args, **kwargs)
    Models the customer object in Visma e-Accounting.

    endpoint /customers

    allowed_methods ['list', 'get', 'create', 'update', 'dlete']

    envelopes {'list': {'class': PaginatedResponse, 'data_attr': 'Data'}}
```

scopes

- ea:sales,
- ea.local:mobile_user

Parameters

- **id** (*uuid.UUID*) – read-only Unique Id provided by eAccounting
- **customer_number** (*str*) – Unique identifier for customer. If none is provided Visma eAccounting API will generate one on creation. Max length: 20 characters.

- **corporate_identity_number** (*str*) – Legal identifier of the customer. If the customer is a private person this field should be their social number. Max lenght = 20 characters.
- **contact_person_email** (*str*) – Email for the contact person at the customer. Max lenght = 255 characters
- **contact_person_mobile** (*str*) – Customer contact person's mobile number
- **contact_person_name** (*str*) – Customer contact person's name
- **contact_person_phone** (*str*) – Customer contact person's phone number
- **currency_code** (*str*) – The code of the currency used when dealing with this customer. Example SEK or EUR default = Currency code in company settings
- **gln** (*str*) – Obsolete Use `edi_gln_number` instead.
- **email_address** (*str*) – Customer email address. Used for sending invoices.
- **invoice_address1** (*str*) – Max length: 50 characters
- **invoice_address2** (*str*) – Max length: 50 characters
- **invoice_postal_code** (*str*) – Max length: 10 characters
- **invoice_city** (*str*) – Max length: 50 characters
- **invoice_country_code** (*str*) – Max length: 2 characters
- **delivery_customer_name** (*str*) – Use if name on deliver differs from customer name. Max length: 100 characters
- **delivery_address1** (*str*) – Use if delivery address differs from customer invoice address. Max length: 50 characters
- **delivery_address2** (*str*) – Use if delivery address differs from customer invoice address. Max length: 50 characters
- **delivery_city** (*str*) – Use if delivery address differs from customer invoice address. Max length: 50 characters
- **delivery_country_code** (*str*) – Use if delivery address differs from customer invoice address. Max length: 2 characters
- **delivery_postal_code** (*str*) – Use if delivery address differs from customer invoice address. Max length: 10 characters
- **delivery_method_id** (*uuid.UUID*) – Reference to the *DeliveryMethod* used.
- **delivery_term_id** (*uuid.UUID*) – Referenct to the *DeliveryTerm* used.
- **pay_to_account_id** (*uuid.UUID*) – read-only Referece to *Account* where payments are registered.
- **name** (*str*) – Customer name. Max length: 50 characters
- **note** (*str*) – Note on customer. Free text. Max length: 4000 characters
- **reverse_charge_on_construction_services** (*bool*) – default=False. If True, VatNumber must be set as well.
- **webshop_customer_number** (*int*) – Reference to customer i webshop.
- **mobile_phone** (*str*) – Customer mobile phone number. Max length: 50 characters

- **telephone** (*str*) – Customer mobile phone number. Max length: 50 characters
- **terms_of_payment_id** (*uuid.UUID*) – Required Reference to the *TermsOfPayment* used.
- **terms_of_payment** (*TermsOfPayment*) – read-only The *TermsOfPayment* used.
- **vat_number** (*str*) – Customers VAT Number. Format: 2 character country code followed by 8-12 numbers. Max length: 20 characters
- **www_address** (*str*) – Customers website. Max length: 255 characters
- **last_invoice_date** (*datetime.datetime*) – read-only Last invoice date.
- **is_private_person** (*bool*) – required default=False
- **discount_percentage** – Customer wide discount specified like 0.9 for 10% discount. Only allows 4 decimals.
- **changed_utc** (*datetime.datetime*) – read-only Last date and time from when a change was made on the customer
- **is_active** (*bool*) – required default=True
- **force_bookkeep_vat** (*bool*) – default=True Not entirely sure what setting this to false will do. Usage Unknown
- **sales_document_language** – Language code for sales documents. Max length: 2 characters
- **edi_gln_number** (*str*) – What is this used for?? Unknown Use
- **electronic_address** (*str*) – Unknown Use Probably have something to do with sending and receiving electronic invoices.
- **electronic_reference** (*str*) – Unknown Use Probably have something to do with sending and receiving electronic invoices.
- **edi_service_deliverer_id** (*str*) – Unknown Use Probably have something to do with sending and receiving electronic invoices.
- **auto_invoice_activation_email_sent_date** (*datetime.datetime*) – Date when electronic invoicing was activated.
- **auto_invoice_registration_request_sent_date** (*datetime.datetime*) – Date when electronic invoicing was requested
- **email_addresses** (*list(str)*) – List of email addresses.

Todo: Need to contact Visma API Team to get proper explanation on:

- force_bookkeep_vat
 - edi_gln_number
 - electronic_address
 - electronic_reference
 - edi_service_deliverer_id
 - Which emails are used for invoicing? all?
-

5.1.14 CustomerInvoiceDraft

class visma.models.CustomerInvoiceDraft (*args, **kwargs)

Represents a Customer Invoice Draft.

endpoint /customerinvoicedrafts

allowed_methods ['list', 'get', 'create', 'update', 'delete']

envelopes {'list': {'class': PaginatedResponse, 'data_attr': 'Data'}}

scope

- ea:sales
- ea:sales_readonly
- ea.local:mobile_user

Parameters

- **id** (*uuid.UUID*) – read-only Unique Id provided by eAccounting
- **customer_id** (*uuid.UUID*) – required Reference to *Customer*
- **created_utc** (*datetime.datetime*) – read-only Creation time. Is automatically set
- **is_credit_invoice** (*bool*) – Indicates if the invoice is a credit invoice. default=False
- **rot_reduced_invoicing_type** (*int*) – Indicates the invoicing type in respect to ROT and RUT. (Sweden) 0 = Normal, 1 = ROT, 2 = RUT. default=0
- **rot_reduced_invoicing_property_name** (*str*) – Unknown use Max length: 40 characters
- **rot_reduced_invoicing_org_number** (*str*) – Unknown use Max length: 11 characters
- **rot_reduced_invoicing_amount** (*number*) – Unknown use Format: 2 decimals, default=0.00
- **rot_reduced_invoicing_automatic_distribution** (*bool*) – Unknown use default=False
- **rot_property_type** (*int*) – Unknown use
- **house_work_other_costs** (*number*) – Unknown use
- **rows** (*list*(*CustomerInvoiceDraftRow*)) – List of *CustomerInvoiceDraftRow* with the details of the invoice. default=list()
- **persons** (*list*(*SalesDocumentRotRutReductionPerson*)) – List of *SalesDocumentRotRutReductionPerson* when using ROT and RUT. Unknown use default=list()
- **your_reference** (*str*) – Customers reference. Max length: 100 characters
- **our_reference** (*str*) – Companys reference. Max length: 100 characters
- **invoice_customer_name** (*str*) – Read-only Customer name on invoice. Max length: 50 characters

- **invoice_address1** (*str*) – Invoice address row 1 Max length: 50 characters
- **invoice_address2** (*str*) – Invoice address row 2 Max length: 50 characters
- **invoice_postal_code** (*str*) – Invoice postal code. Max length: 10 characters
- **invoice_city** (*str*) – Invoice city. Max length: 50 characters
- **invoice_country_code** (*str*) – Invoice country code. Max length: 2 characters, default=SE
- **invoice_currency_code** (*str*) – read-only Invoice currency code
- **delivery_customer_name** (*str*) – Max length: 50 characters
- **delivery_address1** (*str*) – Max length: 50 characters
- **delivery_address2** (*str*) – Max length: 50 characters
- **delivery_postal_code** (*str*) – Max length: 10 characters
- **delivery_city** (*str*) – Max length: 50 characters
- **delivery_country_code** (*str*) – Max length: 2 characters
- **delivery_method_name** (*str*) – Max length: 50 characters
- **delivery_term_name** (*str*) – Max length: 50 characters
- **delivery_method_code** (*str*) – Max length: 20 characters
- **delivery_term_code** (*str*) – Max length: 50 characters
- **eu_third_party** (*bool*) – Indicates if the invoice is subject to rules about EU third party invoicing. default=False
- **customer_is_private_person** (*bool*) – required Indicates if the receiver of the invoice is a private person. default=False
- **reverse_charge_on_construction_services** (*bool*) – read-only Unknown use Need investigation.
- **sales_document_attachments** (*list(uuid.UUID)*) – read-only List of references to attached documents.
- **invoice_date** (*datetime.datetime*) – invoice date
- **delivery_date** (*datetime.datetime*) – delivery date
- **total_amount** (*number*) – read-only Calculated by eAccounting API
- **total_vat_amount** (*number*) – read-only Calculated by eAccounting API
- **total_roundings** (*number*) – read-only Calculated by eAccounting API
- **total_amount_base_currency** (*number*) – read-only Calculated by eAccounting API
- **total_vat_amount_base_currency** (*number*) – read-only Calculated by eAccounting API
- **customer_number** (*str*) – read-only Max length: 16 characters

- **includes_vat** (*bool*) – read-only If true the unit prices on rows include VAT. The value is set upon creation depending whether “Show prices excl. VAT for private individuals” in company’ settings is marked or not

Todo: Contact Visma API Team about rules for ROT and RUT so it can be documented

5.1.15 CustomerLabel

class visma.models.**CustomerLabel** (*args, **kwargs)
 Customer labels. Labels to attach to a customer.

endpoint /customerlabels

allowed_methods ['list', 'create', 'get', 'update', 'delete']

envelopes {'list': {'class': PaginatedResponse, 'data_attr': 'Data'}}

Parameters

- **id** (*uuid.UUID*) – read-only Unique Id provided by eAccounting
- **name** (*str*) – Label name.
- **description** (*str*) – Label description.

5.1.16 DeliveryMethod

class visma.models.**DeliveryMethod** (*args, **kwargs)
 Represents a delivery method in eAccounting.

endpoint /deliverymethods

allowed_methods ['list', 'get']

envelopes {'list': {'class': PaginatedResponse, 'data_attr': 'Data'}}

Parameters

- **id** (*uuid.UUID*) – Unique Id provided by eAccounting.
- **name** (*str*) – Name of delivery method.
- **code** (*str*) – Delivery method code.

5.1.17 DeliveryTerm

class visma.models.**DeliveryTerm** (*args, **kwargs)
 Represents a delivery term in eAccounting.

endpoint /deliveryterms

allowed_methods ['list', 'get']

envelopes {'list': {'class': PaginatedResponse, 'data_attr': 'Data'}}

Parameters

- **id** (*uuid.UUID*) – Unique Id provided by eAccounting.
- **name** (*str*) – Name of delivery term.
- **code** (*str*) – Delivery term code.

5.1.18 FiscalYear

class visma.models.**FiscalYear** (*args, **kwargs)

Represents a fiscal year.

Fiscal years must be created in sequence. For example if you only have 2018 you can't create 2020 until you have created 2019. If you want to create earlier fiscal year they also have to be adjacent to an existing fiscal year

endpoint /fiscalyears

allowed_methods ['list', 'create', 'get']

envelopes {'list': {'class': PaginatedResponse, 'data_attr': 'Data'}}

Parameters

- **id** (*uuid.UUID*) – read-only Unique Id provided by eAccounting
- **start_date** (*datetime.date*) – required
- **end_date** (*datetime.date*) – required
- **is_locked_for_accounting** (*bool*) – read-only Indicates if it is still possible bookkeep on the year.
- **bookkeeping_method** (*int*) – read-only 0 = Invoicing, 1 = Cash, 2 = NoBook-keeping. When posting fiscalyear, previous years bookkeeping method is chosen. ‘

5.1.19 ForeignPaymentCodes

class visma.models.**ForeignPaymentCodes** (*args, **kwargs)

Holds available Foreign Payment Codes.

Parameters

- **id** (*uuid.UUID*) – Unique identifier provided by eAccounting.
- **code** (*int*) – Enum for the payment code.
- **description** (*str*) – Description of the payment code.
- **country_code** (*str*) – Country code.

5.1.20 Project

class visma.models.**Project** (*args, **kwargs)

Represents a Project in eAccounting

Parameters

- **id** (*uuid.UUID*) – read-only Unique Id provided by eAccounting’
- **number** (*str*) – Project number. Max length: 9 characters
- **name** (*str*) – required Project name. Max length: 50 characters

- **start_date** (*datetime.datetime*) – required Project start. Will turn a datetime into date automatically.
- **end_date** (*datetime.datetime*) – Project end.
- **customer_id** (*uuid.UUID*) – Reference to *Customer* for project.
- **customer_name** (*str*) – read-only Customer name depending on customer_id
- **notes** (*str*) – Notes Max length: 500 characters
- **status** (*int*) – Project status. 1 = Ongoing, 2 = Finished
- **modified_utc** (*datetime.datetime*) – Last modified datetime in UTC.

TODO: make custom field to handle the parsing and printing of dates.

5.1.21 OpeningBalances

class visma.models.**OpeningBalances** (*args, **kwargs)

Represents opening balances

Parameters

- **name** (*str*) – Name
- **number** (*int*) – Account number.
- **balance** (*number*) – Account balance.

5.1.22 Supplier

class visma.models.**Supplier** (*args, **kwargs)

Represents a Supplier in eAccounting.

endpoint /suppliers

allowed_methods ['list', 'create', 'get', 'update', 'delete']

envelopes {'list': {'class': PaginatedResponse, 'data_attr': 'Data'}}

Parameters

- **id** (*uuid.UUID*) – read-only Unique Id provided by eAccounting
- **supplier_number** (*str*) – Unique identifier. If not provided, eAccounting will provide one. Max length: 16 characters
- **address1** (*str*) – Supplier address row 1. Max length: 50 characters
- **address2** (*str*) – Supplier address row 1. Max length: 50 characters
- **automatic_payment_service** (*bool*) – Sweden only. Indicates if the supplier is paid by an automatic payment service. Supplier invoices to such suppliers will not be sent to the bank via the bank integration. default=False
- **bank_account_number** (*str*) – Only used in norwegian and danish eAccounting for domestic payments. “Max length: 50 characters”
- **bank_bban** (*str*) – Used on foreign payments to identify a bankaccount together with Bank Code (SupplierBankCode) Format NO: 11 characters, Format DK: 11-14 characters

- **bank_bic** (*str*) – Used on foreign payments to identify a bankaccount together with IBAN (SupplierBankIban) Format: 6 letters followed by 2 or 5 characters (total length 8 or 11)
- **bank_code** (*str*) – Used on foreign payments to identify a bankaccount together with BBAN (SupplierBankBban) Format: 2 letters followed by at least 3 characters
- **bank_country_code** (*str*) – Bank country code. default=Country of the supplier
- **bankgiro_number** (*str*) – Only used in swedish eAccounting, for swedish suppliers. Max length: 10 characters
- **bank_iban** (*str*) – Used on foreign payments to identify a bankaccount together with BIC (SupplierBankBic). Format: 2 letters for country code, 2 control digits, 3 characters for bank identification
- **bank_name** (*str*) – Bank name. Max length: 50 characters
- **city** (*str*) – Supplier city. Max length: 50 characters
- **contact_person_email** (*str*) – Max length: 255 characters
- **contact_person_mobile** (*str*) – Max length: 50 characters
- **contact_person_name** (*str*) – Max length: 50 characters
- **contact_person_phone** (*str*) – Max length: 50 characters
- **corporate_identity_number** (*str*) – Max length: 20 characters
- **country_code** (*str*) – Country code: Max length: 2 characters
- **created_utc** (*datetime.datetime*) – read-only Supplier creation time.
- **currency_code** (*str*) – Currency of the supplier. default=Currency of the user company
- **email_address** (*str*) – Max length: 255 characters
- **mobile_phone** (*str*) – Max length: 255 characters
- **modified_utc** (*datetime.datetime*) – read-only Last modified time.
- **name** (*str*) – Supplier name Max length: 50 characters
- **note** (*str*) – Supplier notes. Max length: 4000 characters
- **plusgiro_number** (*str*) – Only used in swedish eAccounting, for swedish suppliers.
- **postal_code** (*str*) – Max length: 10 characters
- **telephone** (*str*) – Max length: 50 characters
- **terms_of_payment_id** (*uuid.UUID*) – required Reference to [TermsOfPayment](#) used for supplier.
- **www_address** (*str*) – Supplier website. Max length: 255 characters
- **bank_fee_code** (*int*) – Used for foreign payments to determine which party that pays for additional bank fees. 0 = Not set, 1 = SenderPaysAllBankCharges, 2 = RecieverPaysAllBankCharges, 3 = RecieverPaysForeignCosts (Choices taken from app.js source-code) default=0
- **pay_from_bank_account_id** (*uuid.UUID*) – Reference to the [BankAccount](#) is used for foreign payments.

- **foreign_payment_code_id** (*uuid.UUID*) – Reference to ForeignPaymentCode. Used for categorization of foreign purchases (NO and SE only)
- **uses_payment_reference_numbers** (*bool*) – required True if the supplier uses payment reference numbers. (OCR, KID etc.) default=True
- **is_active** (*bool*) – default=True
- **self_employed_without_fixed_address** (*bool*) – default=False

5.1.23 Unit

class visma.models.**Unit** (*args, **kwargs)
Hold available units. Mostly used for articles.

Parameters

- **id** (*uuid.UUID*) – Unique identifier provided by eAccounting.
- **name** (*str*) – Name
- **code** (*str*) – Code
- **abbreviation** (*str*) – Abbreviation.

5.1.24 User

class visma.models.**User** (*args, **kwargs)
Represents a User in eAccounting

Parameters

- **id** (*uuid.UUID*) – Unique identifier.
- **email** (*str*) – Email.
- **first_name** (*str*) – First name.
- **last_name** (*str*) – Last name.
- **is_active** (*bool*) – Indicates if the user is active.
- **is_current_user** (*bool*) – Indicate if the user is the one set up with the current API session.
- **is_consultant** (*bool*) – Indicates if the user is a hired consultant.

5.1.25 VatCode

class visma.models.**VatCode** (*args, **kwargs)
Represents the different VAT Codes used in eAccounting.

endpoint /vatcodes

allowed_methods ['list', 'get']

envelopes {'list': {'class': PaginatedResponse, 'data_attr': 'Data'}}

Parameters

- **id** (*uuid.UUID*) – read-only Unique Id provided by eAccounting
- **code** (*str*) – VAT code
- **description** (*str*) – Description
- **vat_rate** (*number*) – VAT Rate (in percentage??)
- **related_accounts** (*RelatedAccounts*) – A *RelatedAccounts* object that holds the accounts related to this VAT code.

Todo: How is vat rate represented? 0.25 or 25 for 25% VAT?

5.1.26 VatReport

class visma.models.VatReport (*args, **kwargs)
Represents a VAT Report

Parameters

- **id** (*uuid.UUID*) – Unique identifier.
- **name** (*str*) – Name.
- **start_date** (*datetime.datetime*) – Start date of the report.
- **end_date** (*datetime.datetime*) – End date of the report.
- **document_approval_status** (*int*) – Describes the vat reports approval status. 0 = None, 1 = Approved, 2 = Rejected, 3 = ReadyForApproval
- **document_id** (*uuid.UUID*) – Reference to the Document.
- **created_utc** (*datetime.datetime*) – Creatin time in UTC.
- **is_regretted** (*bool*) – Indicates whether the vat report was undone
- **regretted_by_user_id** (*uuid.UUID*) – Reference to the *User* that undid the report.
- **regretted_date** (*datetime.datetime*) – Date the report was undone.
- **modified_utc** (*datetime.datetime*) – Last modified datetime.
- **sent_for_approval_by_user_id** (*uuid.UUID*) – Reference to *User* that send the report for approval.
- **voucher_id** (*uuid.UUID*) – Reference to Voucher
- **total_amount** (*number*) – Predicted vat amount to pay or be refunded.
- **approval_events_history** (*list* (*DocumentApprovalEvent*)) – List of references to *DocumentApprovalEvent* to show the approval history of the VAT Report.

5.1.27 TermsOfPayment

class visma.models.TermsOfPayment (*args, **kwargs)
Describes a term of payment that can be set on customers.
endpoint /termsofpayments

allowed_methods ['list', 'get']

envelopes {'list': {'class': PaginatedResponse, 'data_attr': 'Data'}}

scopes

- ea:sales
- ea:sales_readonly
- ea:purchase
- ea:purchase_readonly
- ea.local:mobile_user

Parameters

- **id** (*uuid.UUID*) – read-only Unique Id provided by eAccounting
- **name** (*str*) – Name of term
- **name_english** (*str*) – English name if term.
- **number_of_days** (*int*) – Number of days until payment.
- **terms_of_payment_type_id** (*int*) – Need more info on how to interpret. Unknown Use
- **terms_of_payment_type_text** (*str*) – Description of term.
- **available_for_sales** (*bool*) – Indicates if term can be used for customer invoices.
- **available_for_purchase** (*bool*) – Indicates if the term can be used for supplier invoices.

Todo: What different types of terms_of_payment_types are there?

5.2 Models without endpoints

5.2.1 CompanyTexts

class visma.models.CompanyTexts (*args, **kwargs)

Collects company wide texts.

Parameters

- **customer_invoice_text_domestic** (*str*) – Text used for domestic invoices. Max length: 180 characters
- **customer_invoice_text_foreign** (*str*) – Text used for foreign invoices. Max length: 180 characters
- **order_text_domestic** (*str*) – Text used for domestic orders. Max length: 180 characters
- **order_text_foreign** (*str*) – Text used for foreign orders. Max length: 180 characters

- **over_due_text_domestic** (*str*) – Text used for domestic over due invoices. Max length: 180 characters
- **over_due_text_foreign** (*str*) – Text used for foreign over due invoices. Max length: 180 characters

5.2.2 CustomerInvoiceDraftRow

class visma.models.**CustomerInvoiceDraftRow** (*args, **kwargs)

Represents a row on a *CustomerInvoiceDraft*

Parameters

- **line_number** (*int*) – required Used to sort the rows. Nothing prevent to have the same line number on several rows. But the order of these will be random. Max=1000
- **article_id** (*uuid.UUID*) – Reference to *Article* Required if is text_row=False,
- **article_number** (*str*) – If not filled eAccounting will get the article number from the specified article ID.
- **is_text_row** (*bool*) – Specifies if row is a text row or article row. default=False
- **text** (*str*) – Article name or Text if is_text_row=True. Max length: 2000
- **unit_price** (*number*) – Use if you want to set a custom price on the article. If not set eAccounting will use the price from the article registry in creation. Format: 2 decimals
- **discount_percentage** (*number*) – Discount on the invoice row. Ex. 10% discount = 0.1 default=0.00
- **quantity** (*number*) – The amount of specified article. Format: 2 decimals
- **work_cost_type** (*int*) – Probably has with ROT and RUT to do. Unknown usage default=0
- **is_work_cost** (*bool*) – Probably has with ROT and RUT to do. Unknown usage default=False
- **work_hours** (*number*) – Probably has with ROT and RUT to do. *Unknown usage*
- **material_costs** (*number*) – Probably has with ROT and RUT to do. *Unknown usage*
- **reversed_construction_services_vat_free** (*bool*) – Probably has with ROT and RUT to do. *Unknown usage* default=False
- **cost_center_item_id1** (*uuid.UUID*) – reference to *CostCenterItem* on invoice row
- **cost_center_item_id2** (*uuid.UUID*) – reference to *CostCenterItem* on invoice row
- **cost_center_item_id3** (*uuid.UUID*) – reference to *CostCenterItem* on invoice row
- **unit_abbreviation** (*str*) – unit abbreviation of the article
- **vat_rate_id** (*str*) – read-only source from *ArticleAccountCoding*
- **unit_name** (*str*) – Name of article unit
- **project_id** (*uuid.UUID*) – reference to *Project*

Todo: Contact Visma API Team about how to handle ROT and RUT.

5.2.3 DocumentApprovalEvent

class visma.models.DocumentApprovalEvent (*args, **kwargs)
 Represents a Document Approval Event

Parameters

- **document_approval_status** (*int*) – Describes the status if the document. 0 = None, 1 = Approved, 2 = Rejected, 3 = ReadyForApproval
- **created_utc** (*datetime.datetime*) – Created time
- **created_by_user_id** (*uuid.UUID*) – Reference to *User* that created the event.

5.2.4 PaginatedResponse

class visma.models.PaginatedResponse (*args, **kwargs)
 Represents the structure of paginated responses on all API endpoints that supports pagination.

Note: As of now this is overridden in model meta class. Had problem getting the attriutes to register on creation

5.2.5 PaginationMetadata

class visma.models.PaginationMetadata (*args, **kwargs)
 Represents the data structure of the meta data for all paginated responses.

Parameters

- **current_page** (*int*) – The current page of the pagination.
- **page_size** (*int*) – Number of object on page.
- **total_number_of_pages** (*int*) – Number of pages
- **total_number_of_results** (*int*) – Total of objects in results.
- **server_time_utc** (*datetime.datetime*) – The servers time serving the request

5.2.6 RelatedAccounts

class visma.models.RelatedAccounts (*args, **kwargs)
 Holds collected information about related accounts.

Parameters

- **account_number1** (*int*) – Account Number 1
- **account_number2** (*int*) – Account Number 2
- **account_number3** (*int*) – Account Number 3

REST-ORM Framework

Part of the client library might be suitable to break out into a separate framework for building ORM clients to different API:s

We try to not couple the functions too hard to the Visma e-Accounting API.

6.1 Inspiration

When looking at the Visma e-Accounting API with the use of OData as filtering query parameters it looks very much like accessing a database. Just over HTTP REST. Normally in Python applications you have an ORM to access a database which usually looks like:

```
>>> Model.objects.all() # Django
```

We have written many integrations where we just have a class that uses request to get the relevant resource and parse the JSON into a dict for further processing. To do this on this entire API would be annoying. Especially since the Visma e-Accounting API uses Pascal casing in their JSON.

So first we looked around for something to handle the Pascal to snake-case serializing and found marshmallow. It did the job well but we had a lot of duplicate code when we wanted to parse data that looked exactly like our Python objects.

So with some inspiration from how Django handles Model creation we made a Metaclass that will use Marshmallow fields to generate a Python object with a similar Marshmallow Schema attached.

6.2 Models

Models inherit from VismaModel.

```
MyClass(VismaModel):  
    id = field.UUID()  
    name = field.String()
```

6.3 Model Meta

To set up how the class interacts with the API we specify the model meta

endpoint Specifies the main endpoint for a class.

allowed_methods Specifies the allowed methods on the class.

envelopes Specifies how to handle enveloped schemas on endpoints.

6.4 Endpoints and methods

We assume for now that one is following normal REST API behaviour. We have some thoughts on how to add specific handling but we have not had time to test other more important features

If you specify an endpoint for example `/customer`, you will get the following behaviour

list accessible via `.all()`. Does a get and get everything from `/customers`

get given an id it will do a GET on `/customers/{id}`

create if no id is set on the model it will POST to `/customers` to create the object on `.save()`

update if id is set on the model it will PUT to `/customers/{id}` to update the object on `.save()`

delete can be called on the object or on the model with an id to issue a DELETE to `/customers/{id}`

Some API endpoint does not have support of all methods so you have to list them in the Meta as a list.

6.5 Enveloped Schemas

When communicating with an API the data sent might be more than the data you want to get or change. For example getting a list of resources on an endpoint that supports pagination you might get meta data like number of pages and current page in a metadata section.

By defining a model for the outer data it is possible to handle this in a simple way by adding the envelope settings in the Meta data of the classes that uses envelope.

```
class PaginatedResponse(VismaModel):
    meta = fields.Nested('PaginationMetadataSchema', data_key='Meta')

class PaginationMetadata(VismaModel):
    current_page = fields.Integer(data_key='CurrentPage')
    page_size = fields.Integer(data_key='PageSize')
    total_number_of_pages = fields.Integer(data_key='TotalNumberOfPages')
    total_number_of_results = fields.Integer(data_key='TotalNumberOfResults')
    server_time_utc = fields.DateTime(data_key='ServerTimeUtc')

class Customer(VismaModel):
    name = fields.String(data_key='Name')

    class Meta:
        endpoint = '/customers'
        allowed_methods = ['list', 'create', 'delete']
        envelopes = {'list': {'class': PaginatedResponse,
                              'data_attr': 'Data'}}
```

This will allow the following data response to return an object of Customer.

```
{'Data': [{'Name': 'Customer-Name'}],  
'Meta': {'CurrentPage': 1,  
         'PageSize': 50,  
         'ServerTimeUtc': '2018-06-21T16:23:13.1083743Z',  
         'TotalNumberOfPages': 1,  
         'TotalNumberOfResults': 37}}
```


CHAPTER 7

Commercial Support and Development

The library is developed by Palmlund Wahlgren Innovative Technology AB, a certified Visma Partner.

[Visma Integration Catalog](#).

We offer software development and integration services to Visma eAccounting (eEkonomi) and can develop you integration or offer support to you team on this library.

Check us out at pwit.se or contact us on info@pwit.se .

CHAPTER 8

Indices and tables

- `genindex`
- `modindex`
- `search`

V

`visma.models`, [13](#)

A

Account (class in visma.models), 13
AccountType (class in visma.models), 14
Article (class in visma.models), 14
ArticleAccountCoding (class in visma.models), 15
ArticleLabel (class in visma.models), 16

B

Bank (class in visma.models), 17
BankAccount (class in visma.models), 17

C

CompanySettings (class in visma.models), 18
CompanyTexts (class in visma.models), 31
CostCenter (class in visma.models), 19
CostCenterItem (class in visma.models), 19
Country (class in visma.models), 20
Currency (class in visma.models), 20
Customer (class in visma.models), 20
CustomerInvoiceDraft (class in visma.models), 23
CustomerInvoiceDraftRow (class in visma.models), 32
CustomerLabel (class in visma.models), 25

D

DeliveryMethod (class in visma.models), 25
DeliveryTerm (class in visma.models), 25
DocumentApprovalEvent (class in visma.models), 33

F

FiscalYear (class in visma.models), 26
ForeignPaymentCodes (class in visma.models), 26

O

OpeningBalances (class in visma.models), 27

P

PaginatedResponse (class in visma.models), 33
PaginationMetadata (class in visma.models), 33
Project (class in visma.models), 26

R

RelatedAccounts (class in visma.models), 33

S

Supplier (class in visma.models), 27

T

TermsOfPayment (class in visma.models), 30

U

Unit (class in visma.models), 29
User (class in visma.models), 29

V

VatCode (class in visma.models), 29
VatReport (class in visma.models), 30
visma.models (module), 13